

RAD's Bid

The Final WPL Project

The project .zip file contains 4 items along with a readme file:

1. OnlineBidding : containing all the front end files (jsp, html, css and js files). It contains the website code along with beans and
2. OnlineBiddingServices : contains all the services, beans and database operations file.
3. sql : contains sql query for setting a schema.
4. Rads Bid.docx : Project Report

The detailed description of all the files is given below:

A. OnlineBidding:

It contains 2 folders in its src folder and a Web Content folder:

1. **Beans:** contains the getters and setters and contains files as follows:
 - a. BidBean.java : setters and getters for every bid
 - b. CartBean.java : setters and getters for cart parameters.
 - c. CartItemBean.java : setters and getters of parameters for every entry in cart.
 - d. PostBean.java : setters and getters for parameters of any post.
 - e. ProductBean.java : setters and getters for parameters of any Product.
 - f. ProductsBean.java : setters and getters for parameters of all Products that enter the shopping cart.
 - g. RegisterBidBean.java : setters and getters for parameters of the bid that gets accepted
 - h. RegisterBidsBean.java : setters and getters for parameters of the final bids that gets accepted and go into shopping cart.
 - i. SearchBean.java : setters and getters for parameters of the bidders and the products.
 - j. UserBean.java : setters and getters for parameters of any user.
2. **Servlets**
 - a. AddToCartServlet.java : servlet controller for adding items to cart
 - b. BidControllerServlet.java : servlet controller for all bids
 - c. CheckBidsServlet.java : servlet controller for all bids on single post
 - d. DeleteCartItemServlet.java : servlet controller for deleting cart items
 - e. EmailCartServlet.java : servlet controller for emailing the user upon purchase of a product and accepting a bid
 - f. LogoutServlet.java : servlet controller for logging out of a user
 - g. PostControllerServlet.java : servlet controller for posting of a new request (product/service)

h. RegisterBidServlet.java	: servlet controller for final bids accepted
i. SearchBiddersServlet.java	: servlet controller for searching bidders
j. SearchProductsServlet.java	: servlet controller for searching products
k. SessionControllerServlet.java	: servlet controller for sessions
l. SignUpControllerServlet.java	: servlet controller for signing up
m. UpdateCountServlet.java	: servlet controller for updating the number of items of a product in cart
n. UpdateProfileServlet.java	: servlet controller for updating profile info
o. ViewBidsServlet.java	: servlet controller for viewing bids
p. ViewCartServlet.java	: servlet controller for viewing cart
q. ViewPostsServlet.java	: servlet controller for viewing posts

Web Content folder contains 2 folders with js and css files. I has the following jsp, jss and css files:

a. AddProduct.jsp	: page for adding a new product
b. Bidding.jsp	: page for viewing all the bids that are posted
c. BiddingDescriptionPage.jsp	: page for posting a new bid
d. checkout1.jsp	: page for checking out
e. Error.jsp	: page displayed when there's an error
f. ExistingPosts.jsp	: page displaying existing posts
g. login-error.jsp	: displays login error
h. LoginPage.jsp	: login page – enters your credentials
i. LogoutPage.jsp	: logs you out of account
j. MainPage.jsp	: main landing page of website
k. ProductDescriptionPage.jsp	: Displays the description of the page
l. ProfilePage.jsp	: displays profile page of user
m. RegistrationPage.jsp	: registration form
n. SearchBidders.jsp	: page displaying results of bidders searched
o. SearchProducts.jsp	: page displaying results of products searched
p. Selling.jsp	: shows the products along with the bids
q. ShoppingCart.jsp	: shows items you want to purchase
r. UpdateProfile.jsp	: update profile page
s. carousel-css.css	: carousel css file
t. cover.css	: cover css file
u. dashboard.css	: dashboard css file
v. signin.css	: css file for sign in
w. theme.css	: css file for theme of the website
x. actb.js	: actb js file
y. sortable.js	: sortable js used for sorted table
z. tablefilter.js	: tablefilter used to add filters to table

B. OnlineBiddingServices

It's got 3 folders in its src folder and they are as follows:

1. Beans: contains the getters and setters and contains files as follows:

- a. BidBean.java : setters and getters for every bid
- b. CartBean.java : setters and getters for cart parameters.
- c. CartItemBean.java : setters and getters of parameters for every entry in cart.
- d. PostBean.java : setters and getters for parameters of any post.
- e. ProductBean.java : setters and getters for parameters of any Product.
- f. ProductsBean.java : setters and getters for parameters of all Products that enter the shopping cart.
- g. RegisterBidBean.java : setters and getters for parameters of the bid that gets accepted
- h. RegisterBidsBean.java : setters and getters for parameters of the final bids that gets accepted and go into shopping cart.
- i. SearchBean.java : setters and getters for parameters of the bidders and the products.
- j. UserBean.java : setters and getters for parameters of any user.

2. Services

- a. AddToCartService.java : adds products to cart
- b. BidServices.java : inserts a new bid in database
- c. DeleteItemCartService.java : deletes item from cart table
- d. DisplayBids.java : displays bids by taking data from db
- e. DisplayPostBids.java : Displays all posts along with their corresponding bids
- f. DisplayServices.java : Displays all the services offered
- g. EmailCartService.java : Emails the user about the order placed
- h. EmailService.java : Emails the user when he/she is registered
- i. LoginServices.java : accepts the credentials and matches up with data in DB
- j. PostServices.java : Inserts new posts into dB
- k. SearchBiddersServices.java : looks up for bidders in dB
- l. SearchProductServices.java : looks up for products in db
- m. SignUpServices.java : registers new users by inserting new info about user in db
- n. UpdateItemCount.java : Updates item count of products in cart in db
- o. UpdateProfile.java : updates profile in db of table users.
- p. ViewCartService.java : displays user cart by fetching data from db

Running the code:

1. Import the OnlineBidding folder in Eclipse Mars 2 as an existing project
2. Import the OnlineBiddingServices folder in Eclipse Mars 2 as an existing project
3. Import the sql file in MySQL benchwork and run the file to create the database schema.
4. In the OnlineBiddingServices folder go to DBO java resource and open DBOperation.
Update the username and password of the sql server database as yours.
5. Run the OnlineBiddingServices on a tomcat 8 server which has its Https port-9443
6. Run the OnlineBidding project on any other instance of tomcat 8 server.